AdafruitVEML6075 Library Documentation

Release 1.0

ladyada

Aug 23, 2019

Contents

CircuitPython library to support VEML6075 UVA & UVB sensor.

CHAPTER 1

Dependencies

This driver depends on:

- Adafruit CircuitPython
- Bus Device

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the Adafruit library and driver bundle.

CHAPTER 2

Usage Example

```
import time
import board
import busio
import adafruit_veml6075

i2c = busio.I2C(board.SCL, board.SDA)

veml = adafruit_veml6075.VEML6075(i2c, integration_time=100)

while True:
    print(veml.uv_index)
    time.sleep(1)
```

CHAPTER 3

Contributing

Contributions are welcome! Please read our Code of Conduct before contributing to help this project stay welcoming.

CHAPTER 4

Building locally

4.1 Zip release files

To build this library locally you'll need to install the circuitpython-build-tools package.

```
python3 -m venv .env
source .env/bin/activate
pip install circuitpython-build-tools
```

Once installed, make sure you are in the virtual environment:

```
source .env/bin/activate
```

Then run the build:

```
circuitpython-build-bundles --filename_prefix adafruit-circuitpython-veml6075 --
↪library_location .
```

4.2 Sphinx documentation

Sphinx is used to build the documentation based on rST files and comments in the code. First, install dependencies (feel free to reuse the virtual environment from above):

```
python3 -m venv .env
source .env/bin/activate
pip install Sphinx sphinx-rtd-theme
```

Now, once you have the virtual environment activated:

```
cd docs
sphinx-build -E -W -b html . _build/html
```

This will output the documentation to `docs/_build/html`. Open the index.html in your browser to view them. It will also (due to -W) error out on any warning like Travis will. This is a good way to locally verify it will pass.

CHAPTER 5

Table of Contents

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/veml6075_simpletest.py

```
import time
import board
import busio
import adafruit_veml6075

i2c = busio.I2C(board.SCL, board.SDA)

veml = adafruit_veml6075.VEML6075(i2c, integration_time=100)

print("Integration time: %d ms" % veml.integration_time)

while True:
    print(veml.uv_index)
    time.sleep(1)
```

5.2 adafruit_veml6075

CircuitPython library to support VEML6075 UVA & UVB sensor.

- Author(s): ladyada

5.2.1 Implementation Notes

Hardware:

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: https://github.com/adafruit/circuitpython/releases
- Adafruit's Bus Device library: https://github.com/adafruit/Adafruit_CircuitPython_BusDevice

class adafruit_veml6075.**VEML6075**(*i2c_bus*, ***, *integration_time=50*, *high_dynamic=True*, *uva_a_coef=2.22*, *uva_b_coef=1.33*, *uvb_c_coef=2.95*, *uvb_d_coef=1.74*, *uva_response=0.001461*, *uvb_response=0.002591*)

Driver base for the VEML6075 UV Light Sensor :param i2c_bus: The `busio.I2C` object to use. This is the only required parameter. :param int integration_time: The integration time you'd like to set initially. Availble options - each in milliseconds: 50, 100, 200, 400, 800. The higher the '_x_' value, the more accurate the reading is (at the cost of less samples per reading). Defaults to 100ms if parameter not passed. To change setting after intialization, use `[veml6075].integration_time = new_it_value`. :param bool high_dynamic: whether to put sensor in 'high dynamic setting' mode :param float uva_a_coef: the UVA visible coefficient :param float uva_b_coef: the UVA IR coefficient :param float uvb_c_coef: the UVB visible coefficient :param float uvb_d_coef: the UVB IR coefficient :param float uva_response: the UVA responsivity :param float uvb_response: the UVA responsivity

integration_time
The amount of time the VEML is sampling data for, in millis. Valid times are 50, 100, 200, 400 or 800ms

uv_index
The calculated UV Index

uva
The calibrated UVA reading, in 'counts' over the sample period

uvb
The calibrated UVB reading, in 'counts' over the sample period

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index